# Client Selection with Rademacher Complexity for Federated Learning

Gabriele Matini, 1934803

**Abstract**

Federated Learning (FL) is a distributed machine learning paradigm where multiple clients collaboratively train a global model without sharing their raw data. This approach enhances privacy and reduces communication costs by transmitting model updates instead of sensitive data. However, FL is often hindered by data heterogeneity, commonly referred to as non-IIDness—where client data distributions vary significantly. This non-identically and independently distributed (non-IID) data leads to challenges in model convergence, fairness, and performance, as the global model must reconcile diverse and potentially conflicting local updates. One method to address the non-IIDness problem in Federated Learning is client selection. In practice, only a subset of clients participate in each training round due to constraints such as limited bandwidth or device availability. This opens the door to designing strategies for selecting the most useful clients—those whose updates contribute most effectively to improving the global model. Clients with richer or more representative data distributions are particularly valuable in this context. Our goal is to identify such clients while fully respecting the privacy-preserving nature of Federated Learning: we cannot directly access client data, nor do we know what an ideal IID distribution would look like. A second objective, is to also provide a useful characterization of a dataset's non-IIDness with regards to its model.

## 1 Direct characterization of clients through their non-IIDness

A fundamental assumption of Machine Learning is the IID-ness, which is constituted by two properties:

1. Independence: each pair of data points must be statistically independent from one another
2. Identical Distribution: every point of the dataset must be drawn from the same underlying probability distribution

In the original 2017 FedAvg algorithm, each of the $K$ clients receives the currenct global model, performs T steps of training locally using SGD, and then sends the updated model weights $\omega_k$ back to the server. The server then performs aggregation of these weights through the following weighted average:

$$w_{\text{global}} = \frac{1}{\sum_{k=1}^{K} n_k} \sum_{k=1}^{K} n_k w_k$$

Where $n_k$ is the number of data samples of client $k$. In the past, it has been demonstrated that non-IIDness has a direct influence on the model updates $\omega_k$, in particular, non-IIDness in FedAvg can be characterized through the weight divergence between the weights calculated by FedAvg and those calculated by normal SGD using an ideal IID dataset [ZLL+18]:

$$weight\ divergence = \frac{\|\omega^{FedAvg} - \omega^{SGD}\|}{\|\omega^{SGD}\|}$$

Such divergence was then demonstrated to be upper bounded by a function of the Earth's Mover Distance between the local probability distributions of the clients and the ideal IID centralized distribution [ZLL+18]. This result tells us that we can describe the impact that non-IIDness will have on the training by analyzing the local client's distribution, for each client. However, privacy is a major concern when it comes to FL, thus such a direct measurement assumes we can access the data directly,

which is not always possible. Moreover, it also assumes that we know how the ideal IID dataset used by normal SGD would look like, in other words, it assumes that we somehow know about the underlying probability distribution.

# 2 Indirect characterization of clients through their non-IIDness

We assume that each client is characterized solely by its local dataset, while the global model is fixed and instantiated identically for all clients. Therefore, at the beginning of each FedAvg training round, each client starts with the same hypothesis function. This will be crucial for measuring non-IIDness, since we can only characterize it through the effects it produces on the client's training.

## 2.1 Clustering of model's updates weights

The first and most obvious indirect measurement that we can use to characterize a client are its updated model weights. In particular, even though clustering model weights does not immediately provide us any information on the quality of the training, like stability or generalization capabilities, given our initial assumptions, it still is a measure of how similar clients' local datasets may be to one another. In practice, if one class of data points is over-represented in two clients, it is reasonable to assume that the clustering will reflect it. On the other hand, if two clients have local datasets that reasonably well reflect the underlying IID distribution, they are likely to cluster together. Clustering could thus be a first step in individuating different and recurrent types of clients within our cluster, allowing us to subsequently offer a more detailed characterization for each group.

## 2.2 Weighted Local Rademacher Complexity as Clients' Characterization

The Rademacher complexity is a foundational tool in statistical learning theory that quantifies the ability of a function class to fit random noise. It provides a measure of the expressiveness—or "capacity"—of a hypothesis class with respect to a given dataset.

**Definition 1 (Empirical Rademacher Complexity)** *Let $S = \{x_1, \ldots, x_n\} \subseteq X$ be a dataset, and let $\sigma_1, \ldots, \sigma_n$ be independent Rademacher random variables (i.e., uniform over $\{-1, 1\}$). Then the empirical Rademacher complexity of a function class $F$ is defined as:*

$$\hat{\mathfrak{R}}_S(F) = \mathbb{E}_{\boldsymbol{\sigma}} \left[ \sup_{f \in F} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(x_i) \right]$$

This expression evaluates how well the function class can align with randomly assigned labels. Intuitively, a high Rademacher complexity implies that the function class is rich enough to overfit to noise—thus generalizing poorly—while a lower value suggests more robust generalization behavior.

In practice, we are usually interested in the generalization behavior of models with respect to a specific loss function, not just their raw outputs. Therefore, we consider the Rademacher complexity of the loss function class $\mathcal{L} = \{\ell(f(x), y) \mid f \in F\}$. This has several advantages:

- It makes the complexity task-specific and tied to generalization error.
- Loss functions are typically bounded or Lipschitz, which helps with theoretical tractability.

In a federated learning (FL) setting, where multiple clients contribute heterogeneous local datasets, it is natural to consider a weighted version of Rademacher complexity across all clients.

Furthermore, since the trained model tends to lie in a low-loss region of the hypothesis space, we can restrict attention to a localized subset, functions with small expected loss, leading to a localized form of the complexity.

**Definition 2 (Weighted Rademacher Complexity)** *Let $X \subseteq \mathbb{R}^D$, $Y \subseteq \mathbb{R}^C$, and let $F$ be a hypothesis class with functions $f : X \to Y$. Let $\ell : Y \times Y \to \mathbb{R}$ be a loss function, and define the associated loss class as:*

$$\mathcal{L} = \{\ell(f(x), y) \mid f \in F\}$$

*Assume there are $K$ clients, each holding a dataset of size $n_k$, with total data $n = \sum_{k=1}^{K} n_k$, and define client weights $p_k = \frac{n_k}{n}$. Then the* Weighted Rademacher Complexity (WRC) *is:*

$$WRC_n(\mathcal{L}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\ell \in \mathcal{L}} \sum_{k=1}^{K} \frac{p_k}{n_k} \sum_{i=1}^{n_k} \sigma_i \ell(f(x_i^{(k)}), y_i^{(k)})\right]$$

## 2.3 Weighted Local Rademacher Complexity

We now introduce the concept of locality in two senses:

1. We evaluate the complexity on the excess loss class $\mathcal{L}^* = \{\ell_f - \ell_{f^*} \mid f \in F\}$, where $f^*$ is the optimal predictor minimizing expected loss. Every $f$ that minimizes the loss "enough", will be near the optimum.
2. We consider only functions in $\mathcal{L}^*$ for which the expected squared excess loss is bounded by a fixed radius $r$.

**Definition 3 (Weighted Local Rademacher Complexity (WLRC))** *Let $f^* \in F$ minimize the expected loss, and define the excess loss class:*

$$\mathcal{L}^* = \{\ell(f(x), y) - \ell(f^*(x), y) \mid f \in F\}$$

*If we consider the localized subset of this class such that:*

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[(\ell(f(x), y) - \ell(f^*(x), y))]^2 \leq r$$

*Then the Weighted Local Rademacher Complexity is defined as:*

$$WLRC_n(\mathcal{L}^*) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\ell \in \mathcal{L}^*} \sum_{k=1}^{K} \frac{p_k}{n_k} \sum_{i=1}^{n_k} \sigma_i \left(\ell(f(x_i^{(k)}), y_i^{(k)}) - \ell(f^*(x_i^{(k)}), y_i^{(k)})\right)\right]$$

This localized complexity allows us to capture the behavior of models near the empirical minimizer, and leads to sharper generalization guarantees.

**Theorem 1 (Excess Risk Bound via WLRC [WLLW22])** *Let $\tilde{f}$ be the function in $F$ minimizing the empirical risk. Then for any $\delta \in (0, 1)$ and for any $G > 1$, with probability at least $1 - \delta$, the following holds:*

$$\mathbb{E}[\ell(\tilde{f}(x), y) - \ell(f^*(x), y)] \leq \frac{800G}{B} r^* + \frac{(16G + 12)B \log(1/\delta)}{n}$$

*where $r^*$ is the fixed point of the function $r \mapsto WLRC_n(\mathcal{L}_r^*)$, and $B$ is a bound on the loss function.*

We now introduce and prove a lemma that serves us to understand more about $r^*$:

**Lemma 1 (Fixed Point Condition for WLRC)** *Let $\mathcal{L}_r^* = \{\ell(f(x), y) - \ell(f^*(x), y) \mid \mathbb{E}_{(x,y)}[(\ell(f(x), y) - \ell(f^*(x), y))^2] \leq r\}$ be the localized excess loss class, and let $WLRC(\mathcal{L}_r^*)$ be the Weighted Local Rademacher Complexity over this class. Then there exists a fixed point $r^*$ such that:*

$$WLRC(\mathcal{L}_r^*) \leq c \cdot r^*$$

The constant c depends on properties of the loss function (e.g., Lipschitz constant) and the hypothesis class. The need for the inequality comes from a result in statistical learning theory, where it is proven that, if we can bound the complexity (WLRC) of the function through a function of the radius, then we can control the generalization error; in other words the excess loss can be tied to the radius. Notice how $r^*$ is tied to the WLRC and to the choice of the "best" cluster, as it would be the lowest $r$ upper bounding the lowest WLRC calculated amongst all the clusters. From the bound it thus follows immediately that the cluster with the lowest calculated WLRC, would be the one with the lowest excess error. We thus propose to use the WLRC as a post-clustering diagnostic to identify the "best" client cluster, the one whose local data distribution most closely aligns with the global learning objective. In fact we can define $r^*$ as:

$$r^* = inf\{r > 0 | WLRC(\mathcal{L}_r^*) \leq c \cdot r\} \tag{1}$$

We next prove that the lemma above: $r^*$ exists and is always well defined. This, in turn, implies proving the sub-root property of the WLRC which is not proven in Wei et al. but just assumed. The sub-root property guarantees the existence of a unique point $r^*$ solution to the fixed-point equation $WLRC(\mathcal{L}_r^*) = r$, which is expressed in [WLLW22].

**Proof 1** *Consider the excess loss class localized by the squared loss radius $r$, and define the function:*

$$\phi(r) := WLRC(\mathcal{L}_r^*)$$

1. *$\phi(r)$ is non-decreasing in $r$: as the radius $r$ gets bigger, the WLRC can only increase or remain equal, since we get further from the optimal $f^*$. In particular it holds that, given an initial set $\mathcal{L}$ of functions and an initial radius $r_1$, and given a particular WLRC score $WLRC(\mathcal{L})$, introducing any new function $\hat{f}$ through a radius $r_2 > r_1$ can have either one of these two effects:*
   (a) *$\hat{f}$ is the new supremum of the WLRC. In this case the WLRC score must necessarily increase or remain the same by supremum monotonicity ($WLRC(\hat{\mathcal{L}}) \geq WLRC(\mathcal{L})$)*
   (b) *$\hat{f}$ is not the new supremum of the WLRC. In this case, the old supremum holds and thus $WLRC(\mathcal{L}) = WLRC(\hat{\mathcal{L}})$.*
2. *$\phi(r)$ is nonnegative: the WLRC is nonnegative by definition. In particular, the supremum of any set of functions here includes always the possibility of choosing $f = f^*$, so we always have at least $WLRC(\mathcal{L}_r^*) = 0$. Of course, $f^*$ will be in the locality of itself.*
3. *$\phi(r)$ is such that, as $r$ increases, $\frac{\phi(r)}{\sqrt{r}}$ decreases or stays constant, i.e. $\phi(r) \leq \hat{c}\sqrt{r}$ for some constant $\hat{c}$.*

   *We are working with the localized excess loss class defined as:*

$$\mathcal{L}_r^* = \left\{ \ell_f - \ell_{f^*} \;\middle|\; \mathbb{E}_{(x,y)} \left[ \left( \ell(f(x), y) - \ell(f^*(x), y) \right)^2 \right] \leq r \right\}$$

   *Let $f \in \mathcal{L}_r^*$. Then, by the definition of the class:*

$$\mathbb{E}_{(x,y)} \left[ \left( \ell(f(x), y) - \ell(f^*(x), y) \right)^2 \right] \leq r$$

   *We now use the assumption that the loss function $\ell(\cdot, y)$ is L-Lipschitz in its first argument. Therefore:*

$$|\ell(f(x), y) - \ell(f^*(x), y)| \leq L \cdot |f(x) - f^*(x)|$$

   *Squaring both sides, we obtain:*

$$\left( \ell(f(x), y) - \ell(f^*(x), y) \right)^2 \leq L^2 \cdot \left( f(x) - f^*(x) \right)^2$$

   *Now take expectation over $(x, y) \sim \mathcal{D}$:*

$$\mathbb{E}_{(x,y)} \left[ \left( \ell(f(x), y) - \ell(f^*(x), y) \right)^2 \right] \leq L^2 \cdot \mathbb{E}_x \left[ \left( f(x) - f^*(x) \right)^2 \right]$$

   *Rewriting this:*

$$\mathbb{E}_x \left[ \left( f(x) - f^*(x) \right)^2 \right] \geq \frac{1}{L^2} \cdot \mathbb{E}_{(x,y)} \left[ \left( \ell(f(x), y) - \ell(f^*(x), y) \right)^2 \right]$$

   *Now, in practice we use a radius $r'$ such that $|f(x) - f^*(x)|^2 < r', \forall f \in \mathcal{F}$, where $\mathcal{F}$ contains the functions $f$ that help define the local subset $\mathcal{L}_r^*$. If $r < r'$ we just consider then $r = r'$, and the previous arguments apply the same.*

   *By the definition of $f \in \mathcal{L}_r^*$, the right-hand side is at most $r$ too, so we can say that:*

$$\mathbb{E}_x \left[ \left( f(x) - f^*(x) \right)^2 \right] \leq \frac{r}{L^2}$$

*Hence, we have shown that any function $f \in F$ lies within an $L_2(\mathcal{D}_x)$ ball of radius $\sqrt{r}/L$ centered at $f^*$:*

$$\|f - f^*\|_{L_2(\mathcal{D}_x)} \leq \frac{\sqrt{r}}{L}$$

*Thus, the localized class $\mathcal{L}_r^*$ is contained within a function class with $L_2(\mathcal{D})$ norm bounded by $\sqrt{r}/L$. We then apply the Talagrand's contraction principle, which states that, for all $f \in F$, if the loss is L-Lipschitz, then $WLRC(\mathcal{L}_r^*) \leq L \cdot WLRC(F)$, because $\mathcal{L}_r^*$ is created with functions resulting from the composition $\ell \circ F$. Next, from the empirical process theory (Dudley entropy integral bound), we derive another bound that ties $WLRC(F)$ directly to some function of the radius. Let $\{X_\theta : \theta \in \mathbb{T}\}$ be a zero-mean, sub-gaussian process with $L_2$ metric on $\mathbb{T}$. Then*

$$\mathbb{E}[sup_{\theta,\theta' \in \mathbb{T}}(X_\theta - X_{\theta'})] \leq C \int_0^D \sqrt{logN(\mu, \mathbb{T})}\ d\mu$$

*. Notice how our WLRC is already a zero-mean, subgaussian process, since the $\sigma_i$ have an expected value of 0, and they describe a Bernoulli process, which is subgaussian. $N(\mu, \mathbb{T})$ is defined as the covering number of $\mathbb{T}$, which is the smallest number of balls of radius $\mu$ needed to cover all of $\mathbb{T}$. In our case we define $\mathbb{T} = F$, and $X_f - X_{f^*} = \frac{1}{n}\sum_{i=1}^n \sigma_i(f(x) - f^*(x))$. Then, at the left hand side of the disequation we get the definition of the WLRC:*

$$\mathbb{E}[sup_{f,f' \in F}(X_f - X_{f^*})] \leq C \int_0^{\frac{\sqrt{r}}{L}} \sqrt{logN(\mu, F)} d\mu$$

*, Where $logN(\mu, F)$ is called metric entropy. Now, to bound the term inside the integral the metric entropy itself needs to be finite, and it is, since all of the considered functions are contained in a finite space defined by the radius $\frac{\sqrt{r}}{L}$. $F$ already has finite metric entropy since $\|f(x) - f^*(x)\| \leq \frac{\sqrt{r}}{L} \forall f \in F$, so we can have under standard results:*

$$logN(\mu, F) \leq C \cdot (\frac{\sqrt{r}/L}{\mu})^2 \implies \int_0^{\frac{\sqrt{r}}{L}} \sqrt{logN(\mu, F)} d\mu \leq C \int_0^{\frac{\sqrt{r}}{L}} \frac{\sqrt{r}/L}{\mu} d\mu$$

*Truncating and solving the second integral at $\epsilon_0$ instead of 0 (we need to because at 0 it diverges), we get, by assimilating all constants to $\hat{c}$:*

$$WLRC(F) \leq \hat{c}\frac{\sqrt{r}}{L}$$

*Recalling the Talagrand's inequality we can write:*

$$\frac{WLRC(\mathcal{L}_r^*)}{L} \leq WLRC(F) \leq \hat{c}\frac{\sqrt{r}}{L} \implies \frac{WLRC(\mathcal{L}_r^*)}{\sqrt{r}} = \frac{\phi(r)}{\sqrt{r}} \leq \hat{c}$$

*Therefore,*

$$\frac{\phi(r)}{\sqrt{r}} \leq \hat{c} \quad \textit{and is non-increasing in } r. \ \square$$

This guarantees that the fixed point equation cited before has some solution $r^*$. Once $r^*$ is found, it can be plugged into some localized generalization bound of the form:

$$P(\ell_{\hat{f}} - \ell_{f^*}) \leq C_1 r^* + C_2 \frac{\log(1/\delta)}{n}$$

where $C_1$ depends on the boundedness of the loss function. In the WLRC case, $C_1 = 800G/B$ corresponds to the constant derived in the paper.

We finally now do a summary of why we have proved that we can use the WLRC to assess client's "goodness" of data distribution.

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two clusters with corresponding localized complexity functions $\phi_1(r) = \text{WLRC}_1(\mathcal{L}_r^*)$ and $\phi_2(r) = \text{WLRC}_2(\mathcal{L}_r^*)$, respectively. Assume that both clusters share the same hypothesis space $F$

and that training is initialized at the same function $f_0 \in F$. These conditions ensure that the definitions of the localized excess loss class $\mathcal{L}_r^*$ and the fixed point equation $\phi(r^*) \leq c \cdot r^*$ are comparable across clusters.

Suppose we observe:

$$\text{WLRC}_1(\mathcal{L}_r^*) < \text{WLRC}_2(\mathcal{L}_r^*) \quad \text{for } r_1^* \text{ and } r_2^*$$

Since each $\phi_i(r)$ is non-decreasing in $r$ (i.e., larger radii allow more functions and hence potentially higher supremum), it must be that the fixed point radius $r_1^*$ of cluster $\mathcal{C}_1$ satisfies:

$$r_1^* < r_2^*$$

To see why, suppose by contradiction that $r_1^* > r_2^*$. Then by monotonicity:

$$c \cdot r_1^* \geq WLRC_1(\mathcal{L}_r^*) \geq WLRC_2(\mathcal{L}_r^*)$$

then we can have either

$$WLRC_1(\mathcal{L}_r^*) \geq c \cdot r_2^* \geq WLRC_2(\mathcal{L}_r^*)$$

which contradicts the fact that

$$\text{WLRC}_1(\mathcal{L}_r^*) < \text{WLRC}_2(\mathcal{L}_r^*)$$

was observed, or we can have

$$c \cdot r_1^* > c \cdot r_2^* \geq WLRC_2(\mathcal{L}_r^*) > WLRC_1(\mathcal{L}_r^*)$$

which contradicts the fact that $r_1^*$ is the result of the fixed point equation. Hence, the only possibility is $r_1^* < r_2^*$.

This implies that the generalization bound:

$$P(\ell_{\hat{f}} - \ell_{f^*}) \leq C_1 r^* + C_2 \frac{\log(1/\delta)}{n}$$

is tighter for cluster $\mathcal{C}_1$, and thus a lower WLRC score corresponds directly to lower expected excess error. Therefore, WLRC can be used as a valid criterion to select the cluster with the best generalization properties.

In conclusion, under the assumption of shared hypothesis class and initialization, a lower WLRC implies a lower fixed point radius $r^*$, which leads directly to a tighter generalization bound. This justifies using the WLRC as a formal post-clustering diagnostic to identify the cluster with the strongest generalization performance.

# 3 Implementation of the procedure

We now present how the procedure would work in practice under the FL-FedAvg paradigm.

## 3.1 The clustering

We propose the clustering at the end-point of a FedAvg step, so when the local training is over the clients should send the server their local updated weights to be server, the update weights can then be clustered. This step is straightforward enough, but careful consideration must be employed in choosing the amount of clusters. Another key factor is initialization: every model on every client should be initialized in the same way (same initial weights) and should be, of course, the same.

## 3.2 WLRC estimation

It is not possible to directly compute the WLRC, since in practice we do not know $f^*$. Fortunately there are methods to estimate for each client the local rademacher complexity. These results can then be aggregated at the server level to compute an estimation of said quantity. The estimation involves two steps: calculating a local approximation of the WLRC, and aggregating the values at the server level. [WLLW22] proposes this method to estimate the WLRC.

**Algorithm 1** Approximating WLRC score

**Input:** $w$: model parameters, $k$: local iterations, $B$: mini-batch size, $C$: number of classes, $Q$: number of times the Rademacher variables are sampled

**Note:** This is the local WLRC estimation procedure

at the end of the FedAvg local training step do:
**for** each batch $\{(x_i, y_i)\}_{i=1}^{B}$ **do**
  $k \leftarrow 0$
  **for** $q = 1, \ldots, Q$ **do**
    Sample Rademacher variables $\{\sigma_{ic}\}_{c=1,\ldots,C}^{i=1,\ldots,B}$
    $k \leftarrow k + \left| \frac{1}{BC} \sum_{i=1}^{B} \sum_{c=1}^{C} \sigma_{ic} f_c(x_i) \right|$
  **end for**
  $k \leftarrow \frac{k}{Q}$
  $WLRC = k$ Send $WLRC$ to server.
**end for**

**Algorithm 2** Aggregating the results at the server level

**Input:** $K_i$: number of clients in cluster i

**Note:** This is the server aggregation procedure

$WLRC_{scores} = \{\}$
**for** each cluster $C_i$ **do**
  $WLRC_i \leftarrow 0$
  **for** each client $k$ in cluster $C_i$ **do**
    $WLRC_i \leftarrow WLRC_i + LocalWLRC(k)$
  **end for**$WLRC_{scores} \leftarrow WLRC_{scores} \cup \{\frac{WLRC_i}{|C_i|}\}$
**end for**
return $WLRC_{scores}$

Note that for simpler models (shallow models) the estimation of the WLRC is simpler and probably provides better results, since [WLLW22] provides a result that lets us use SVD decomposition on the learning parameters $W$ to estimate the WLRC.

# References


[WLLW22] Bojian Wei, Jian Li, Yong Liu, and Weiping Wang. Non-iid federated learning with sharper risk bound. *IEEE Transactions on Neural Networks and Learning Systems*, 35:6906–6917, 2022.

[ZLL+18] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.